WEST

Generate Collection

L25: Entry 1 of 1 File: USPT Aug 6, 2002

DOCUMENT-IDENTIFIER: US 6430718 B1
TITLE: Architecture, circuitry and method for testing one or more integrated circuits and/or receiving test information therefrom

Detailed Description Text (15):
During normal functional operation of integrated circuit 50, TAP controller 70 is in state 82. In this state, the data on the input conductor of input scan device (DATA IN) is connected to DATA OUT sent into the core logic. More specifically, in a normal functional operation, whatever is input to the input pad will be forwarded into core logic, and whatever is output from the core logic will be forwarded to the output pad. Connection of DATA IN and DATA OUT is effectuated through control of TAP CNTL 2 signal. The TAP CNTL 2 signal as well as TAP CNTL 1, TAP CLK 1, and TAP CLK 2, are derived from a master clock signal which is a buffered version of the TCK signal defined in the JTAG standard. Upon commencement of a test operation, TAP controller 70 will first transition to test-idle state 84. Transitions 86 occur between states 82 and 84 depending on whether normal operation or test operation occurs. However, once in an appropriate state, TAP controller 70 will remain in that state unless a transition 86 occurs. If a scan test is to be carried out, the TAP controller 70 will, on the next cycle of the master clock, enter selective-scan state 88. On the next clock cycle of the master clock, TAP controller 70 will enter a capture state 90. When in state 90, TAP CNTL 1 signal will be such that the input pin will be connected directly to the input of the capture latch 46. On the next transition of the master clock, a shift state 92 occurs. In this state TAP controller 70 produces a plurality of clock cycles, such as TAP CLK 1 while maintaining TAP CNTL 1 at a level such that the SERIAL TEST IN signal appears at the input of capture latch 46. Each transition of TAP CLK 1 will serially shift the captured ATE test data into the test next scan device. The number of shifts N is dependent upon the number of input (or output) boundary scan cells and/or the number of scan elements within the internal scan chain. Thus, the number of clock cycles necessary to shift the test vectors from cells 52 into the core logic (or into the scan elements 53) and from the core logic to scan elements 53 (or boundary cells 54) is represented as 2N+2M, where N represents the number of boundary scan cells and M represents the number of scan elements. Hence, for 2N+2M cycles of the master clock, shift state 92 is maintained. On the next cycle of the master clock or sometime thereafter, an update state 94 is entered. While in the update state 94, TAP controller 70 will cause the output of update latch 48 to be updated by generating a clock cycle of TAP CLK 2. The test vector data bit or field of bits, on the input of update latch 48 will therefore appear on the output of latch 48, and consequently to the output pin as DATA OUT, or to the core logic as DATA IN. In the input boundary scan example, all of the DATA IN signals will have been updated and thereafter sent into integrated circuit 14, as well as various integrated circuits serially connected thereto.

Current US Original Classification (1):
714/727

Current US Cross Reference Classification (1):
714/729

L4: Entry 10 of 12 File: TDBD Aug 1, 1989

TDB-ACC-NO: NA8908427

DISCLOSURE TITLE: Method of Identifying Non-Repairable Fail Patterns

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, August 1989, US

VOLUME NUMBER: 32
ISSUE NUMBER: 3A
PAGE NUMBER: 427 - 428

PUBLICATION-DATE: August 1, 1989 (19890801)

CROSS REFERENCE: 0018-8689-32-3A-427

DISCLOSURE TEXT:

- Random-access memory (RAM) arrays having a finite number of redundant lines in each of two dimensions are tested, and the location of failing cells (fails) is stored. Unfixable distributions of fails are identified early in repair calculations, thus saving calculation and test time. - Algorithms for repairing two dimensionally redundant arrays are usually comprised of two discrete portions, a "must fix" and a "sparse" phase. Lines which contain a number of fails which exceeds the number of redundant lines in the orthogonal dimension are placed in the must fix category and each such line utilizes a redundant line. - After the identification of all the must fix lines, any remaining fails are classified as sparse fails and their location pattern is compressed into a smaller memory of the tester to speed access for the repair calculation. Algorithms to resolve the sparse fails typically employ an exhaustive search of all possible combinations of line replacements and can be time consuming, especially for the non-repairable patterns. - For an array with sparse fails to be repairable, the matrix created by compressing the fails remaining after all must-fix lines and fails have been removed is bounded by: 1. The number of unique row fails R must be less than or equal to the number of available redundant rows M times the number of redundant columns N plus one, $R <= M \times (N + 1)$. 2. The number of unique column fails $C <= N \times (M + 1)$. 3. The total number of fails $T <= 2 \times (M + N)$. - The fact that a sparse fail pattern meets the parameters R, C, and T does not guarantee that repair is possible. A fourth test is applied to arrays having sparse fails to further reduce the number of non- repairable patterns that the sparse algorithm must process. This fourth test detects patterns which contain a total number of fails, each fail having unique x and y coordinates designated U fails, which exceeds the number of spare lines remaining. - Patterns comprised of only U fails can be detected during the must fix process. At the completion of counting the fails along each line, a latch is set if the fail count is greater than one and the line has not been flagged as a must fix line. If the latch has not been set when the must fix process is ended, any fails remaining are U fails. Next, when the total number of U fails counted as the fails are transferred to the compressed matrix exceeds the number of remaining spare lines, a non repairable pattern is identified and the sparse algorithm is bypassed.